WhistleBlower: Towards A Decentralized and Open Platform for Spotting Fake News

Gowri Ramachandran, Daniel Nemeth
Viterbi School of Engineering
University of Southern California
Los Angeles, USA
{gsramach, dnemeth}@usc.edu

David Neville, Dimitrii Zhelezov,
Ahmet Yalcin, and Oliver Fohrmann
Helix Foundation
Berlin, Germany
{dn, dz, ay, of}@hlx.ai

Bhaskar Krishnamachari
Viterbi School of Engineering
University of Southern California
Los Angeles, USA
{bkrishna}@usc.edu

***Abstract*—The vast majority of the population is consuming news from various digital sources, including social networking applications such as Twitter and Facebook and other online digital platforms. Such Internet platforms provide malicious entities an opportunity to spread fake news and hoaxes to mislead the population. Besides, Internet users may start to form an opinion and make certain personal or business decisions based on misinformation, leading to undesirable consequences. This paper introduces WhistleBlower, a decentralized and open platform based on the blockchain and distributed ledger technology (DLT) for spotting fake news. The key components of WhistleBlower include a *fake news processing engine* powered by Artificial Intelligence (AI)/Machine Learning (ML) algorithms, a *verifiable computation engine*, and *a token-curated registry* (TCR).**

WhistleBlower allows the community members to participate in the fake news identification process by running the *fake news detection algorithm* on their nodes, which would then be validated by *a verifiable computation engine* to ensure that the public nodes executed the computation honestly and correctly. Whenever a news feed is submitted to WhistleBlower for fake news assessment, it issues a genuineness score, which can then be posted along with the news article to let the newsreaders gauge its legitimacy. However, the genuineness score's accuracy depends on the machine learning model's effectiveness that processes the news item. To improve the machine learning algorithm's reliability, we introduce *a Token-curated registry*, which enables the public and community members to challenge the algorithm used to estimate the genuineness score. TCR lets the community curate fake news detection algorithms by providing feedback to the ML/AI algorithm developers through the token-curated content moderation process. WhistleBlower is the first open and democratic fake news assessment platform that combines ML/AI, verifiable computation, and TCR to the best of our knowledge.

***Index Terms*—Fake news, Blockchain, Verifiable computation, Token curated registry, TCR, WhistleBlower**

I. Introduction

The digital platforms have created a massive impact in the last decade, resulting in the wide-spread adoption of online services, including social media platforms such as Facebook and Twitter and multiple online news consumption platforms. Although these digital platforms offer benefits to the community, it also allows the malicious parties to spread false information to a broad array of people with minimal effort. Real-world incidents [1]–[3] show that fake news could impact people's behavior, and it may make people form opinions based on incorrect information. It is, therefore, essential to develop solutions to identify and prevent the spread of fake news.

Spotting fake news involves the processing of news sources and their credibility. On the one hand, several solutions have been proposed in the literature to detect fake news using machine learning (ML) and artificial intelligence (AI) algorithms [4]–[6]. On the other hand, the blockchain technology has been considered to detect fake news using smart contracts [7], [8]. The former approaches present various algorithms and assume the centralized entity that runs the algorithms are honest and unbiased. The latter implies that smart contracts are capable of processing computationally-intensive AI algorithms inside the blockchain. *A decentralized and community-driven platform for fake news detection remains an open problem, which is the focus of this work.*

The blockchain and DLT platforms such as BitCoin and Ethereum have disrupted the financial sector. Other domains, including the Internet of Things [9], [10], are also considering blockchain-based frameworks to provide trust guarantees for the application stakeholders. These efforts show that the blockchain technology offers an elegant solution to create decentralized application architectures capable of operating in a multi-stakeholder environment while providing transparency. Such benefits motivate us to apply blockchain and DLT to spot fake news.

This work introduces WhistleBlower, a decentralized, democratic, and community-driven platform for detecting fake news. Our framework consists of fake news processing algorithms, a verifiable computation engine, and a token-curated registry. WhistleBlower

- Relies on AI or ML algorithms to assess the news item's genuineness based on the source.
- Uses a verifiable computation framework to reliably offload the execution of fake news detection algorithm to the compute nodes contributed by the community members, wherein the verifiable computation framework is used to ensure the correctness of the computation performed by the public nodes.
- Includes a token-curated registry to help the news consumers challenge the article's genuineness, thereby providing feedback to the developers of ML and AI algorithms.

WhistleBlower's architecture, design issues, security, and trust analysis are presented. The effectiveness and the feasibility of verifiable computation are demonstrated through a novel verifiable Python (vPython). Besides, TCR's importance and how the community can effectively curate ML/AI algorithms and earn tokens in WhistleBlower is elucidated through TCR simulations. WhistleBlower is the first decentralized fake news detection platform that combines ML/AI, verifiable computation, and TCR to the best of our knowledge.

II. Fake News and Its Societal Implications

Lazer*et al.* [11] define Fake News as "...fabricated information that mimics news media content in form but not in organizational process or intent. Fake-news outlets, in turn, lack the news media's editorial norms and processes for ensuring the accuracy and credibility of the information. Fake news overlaps with other information disorders, such as misinformation (false or misleading information) and disinformation (false information that is purposely spread to deceive people)". Although several other definitions exist in the literature, we believe this definition captures the modalities of fake news, including the roles of misinformation and disinformation.

A. Societal Implications of Fake News

Given the global and wide-spread adoption of the Internet and social media platforms, a vast population is exposed to tens to hundreds of media contents and news feeds daily. Under such circumstances, people start to form opinions about the world, a country, an individual, an organization, and other subjects, including religion and politics. Remember that modern-day digital platforms, including social media sites, allow any individual to easily create, share, and spread information with other parties in their network. Unfortunately, such an operational model of the social media platforms allows for all kinds of information to easily spread to thousands of individuals rapidly. Note that such platforms can make positive impacts, including the possibility of creating a connected society, but some malicious actors are misusing them for various reasons. We provide the negative consequences of fake news through a few examples from the literature:

- In December 2016, a government diplomat responded with strict defense measures [1], including nuclear threat after reading a fake news[1].
- Another study by Grinberg *et al.* [2], reported that fake news influenced an election process.
- Sharma *et al.* [3] presents how fake news is spreading in the wake of CoVID-19 (SARS-CoV2), and explains how people are exposed to mixed information, including on the topic of social distancing and its effectiveness.

The literature [12], [13] presents a comprehensive survey of fake news and discusses approaches to prevent fake news spread. Such studies and the real-world examples show that fake news can influence the people who can make decisions while enabling malicious actors to divide communities.

How Fake News spreads? Social networking sites and digital media platforms such as Wikipedia and Medium allow anyone to easily create and share information with people in their network, spreading to many people, for example, through tweets and retweets [13].

III. Related Work and Gap

The literature presents various approaches to detect fake news. Such approaches classify the fake news analysis schemes into three categories based on the *style - how the news article is written*, *propagation - how it spreads*, and *user - who is sharing or spreading* [12]. These analyses present ML and AI algorithms [4]–[6] to detect fake news based on a model that was trained using the data collected from social media sites in combination with fact-checking tools such as ClaimBuster [14] and factcheck.org[2]. Such approaches either:

- Present an algorithmic solution that is capable of predicting the genuineness of the news article using deep learning and other advanced ML and AI techniques, or
- Present tools [14] to help the users self-check the legitimacy of the news article. Moreover, such tools are owned and managed by a single stakeholder following a centralized architecture, which is susceptible to a single point of failure.

A. Gap

Contemporary ML and AI-based solutions are promising for fake news detection, but such solutions,

- Do not present any approach to prevent a single stakeholder from controlling and managing the fake news detection process.
- Do not present methods to involve community members in the fake news detection and curation processes.
- Do not offer any incentives to encourage community participation.

In this work, we focus on developing a decentralized and open platform for fake news detection by involving the community members through blockchain/DLT and incentive schemes. Our work is complementary to existing algorithmic approaches presented in the literature; it provides the fake news algorithm developers an opportunity to apply their solutions in a practical setting.

IV. WhistleBlower: A Decentralized and Democratic Platform for Spotting Fake News

Figure 1 shows the architecture and the critical building blocks of WhistleBlower, which we will describe below.

A. System Model and Assumptions

We model WhistleBlower as follows:

Clients. There are $\mathcal{U}$ clients interested in verifying the genuineness of news articles. Each client, $\mathcal{U}_i$, submits a news article when sending a verification request to Smart Contract (discussed below). For simplicity, we assume that the news article contents are provided in the form of a text file.

[1] https://www.nytimes.com/2016/12/24/world/asia/pakistan-israel-khawaja-asif-fake-news-nuclear.html?$_r = 0$

[2] https://www.factcheck.org/

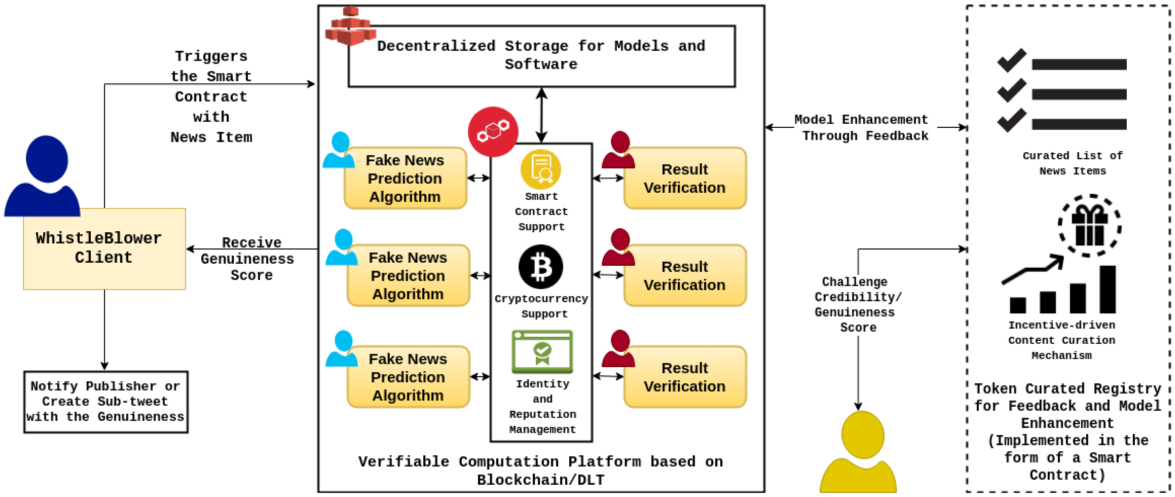
Triggers the Smart Contract with News Item
Decentralized Storage for Models and Software
Model Enhancement Through Feedback
Curated List of News Items
WhistleBlower Client
Receive Genuineness Score
Fake News Prediction Algorithm
Smart Contract Support
Result Verification
Fake News Prediction Algorithm
Cryptocurrency Support
Result Verification
Challenge Credibility/ Genuineness Score
Incentive-driven Content Curation Mechanism
Notify Publisher or Create Sub-tweet with the Genuineness
Fake News Prediction Algorithm
Identity and Reputation Management
Result Verification
Token Curated Registry for Feedback and Model Enhancement (Implemented in the form of a Smart Contract)
Verifiable Computation Platform based on Blockchain/DLT

Fig. 1. The Architecture of WhistleBlower.

Algorithms. We consider a set of ML and AI algorithms, $\mathcal{A}$, which can calculate the genuineness score of news articles. We assume that the algorithm developers register their algorithms on WhistleBlower. At the time of registration, the algorithm developers make a deposit, $\mathcal{A}_d$.

Computation Software Package. The computation software package consists of the client's news article, and the pointer to the code for the genuineness calculation algorithm, $\mathcal{A}_i$, selected by the smart contract.

Verification Software Package. The verification software package consists of the client's news article (input), the pointer to the code for the genuineness score calculation algorithm, $\mathcal{A}_i$, genuineness score (output), and the run-time and stack traces as the proof of execution.

Decentralized Storage. WhistleBlower is required to store the code and models associated with ML and AI algorithms, computation software packages, and verification software packages. The blockchain ledgers are not suitable for storing hundreds of megabytes of data. Therefore, we assume that the algorithm developers, solvers, and verifiers would submit their code and packages to decentralized storage and submit only the pointers to the storage to the smart contract.

Verifiable Computation. We consider a verifiable computation platform, $\mathcal{V}$, which is open for any public entity with computation resources. At its core, there is a smart contract, $\mathcal{C}$, which handles the requests of clients ($\mathcal{U}$), and it selects m nodes to run the fake news detection algorithm. In this work, we assume $m = 2$ for simplicity. Here, we assume that the smart contract, $\mathcal{C}$, selects an algorithm for genuineness score calculation from the algorithms' list.

Solvers. Within $\mathcal{V}$, there is a set of solver nodes, $\mathcal{V}_i^S$, whose job is to receive a computation software package from a contract, $\mathcal{C}$, execute it, and submit the results back to the contract $\mathcal{C}$. At the time of execution, each solver node, $\mathcal{V}_i^S$, uses the news article provided by the client, $\mathcal{U}_i$, as an input to the fake news assessment algorithm, and it produces the genuineness score. Here, the public solver nodes assess the genuineness of the article, wherein the solver nodes may submit fake results without running the computation. It is, therefore, essential to verify the correctness of the computation. Thus, each solver node collects the run-time and stack traces, $\mathcal{T}$, as a proof for executing the computation, which is included in the verification software package.

Provers (or Verifiers). Within $\mathcal{V}$, there is a set of verifier nodes, $\mathcal{V}_i^P$, whose role is to verify the correctness of the computation performed by the solver nodes. The smart contract, $\mathcal{C}$, selects a computation node for the verification. Upon selecting a node for the verification, the smart contract, $\mathcal{C}$, dispatches the verification software package. Here, the verifier node validates the correctness by comparing the outputs, run-time, and stack traces.

- When the stack trace matches, the verifier notifies the smart contract that the results are correct, at which point, the smart contract notify the result back to the client. Besides, the solver nodes and verifier nodes are rewarded for their participation.
- When the stack traces don't match, the verifier informs the mismatch to the smart contract, which then penalizes both the solver nodes (we assume there are only two solver nodes), and restarts the entire process by selecting a new set of solver nodes.

We assume that the smart contract can select a node for solving and verification roles randomly for simplicity. In our ongoing work, we are developing an approach to elect solver and verification nodes securely.

Token-curated Registry. WhistleBlower consists of a token-curated registry (TCR), a community-driven content curation framework based on the blockchain technology and smart contract. We use the TCR to curate the ML and AI algorithms that calculate the genuineness score. All the algorithms that are registered on the WhistleBlower smart contract are automatically added to the TCR listing. WhistleBlower smart contract selects algorithms randomly from the list of algorithms to calculate the genuineness score for a news article submitted by a client, $\mathcal{U}_i$. When a newsreader comes across an article that was verified using WhistleBlower, and she/he notices that the score is incorrect, then she/he can challenge the algorithm through TCR.

Challengers. These are newsreaders who are not satisfied with the news item's genuineness score. Here, the newsreader may feel that the genuineness score is too low or too high based on his awareness and the known facts. Under such circumstances, the newsreader may initiate a challenge by questioning the algorithm's correctness that calculated the genuineness score.

Voters. When a challenge is created, the community members can vote either in favor or against the challenge. Those who participate in the challenge are called voters, and their role is critical to maintaining the quality of WhistleBlower and its algorithms.

B. WhistleBlower Client

It is a client-side software running outside the blockchain. The client allows the community members to post a news item, including its contents, source, and other metadata that may be useful for the ML/AI algorithm to process the data. The client request is posted to the WhistleBlower smart contract, which is running inside the blockchain. Here, we assume that the contents are stored in a decentralized storage and only the pointer to the package is sent to the smart contract.

After processing the news item, the WhistleBlower core returns the result, including the genuineness score and the algorithm identifier. The client can then post the result as a sub-tweet to the tweets that spread the fake news. Besides, the digital content distributors can also attach the genuineness score to their news articles to provide a guideline to the newsreaders. Note that such a model forces the content distributors to post only news items with a high genuineness score. However, the accuracy of the genuineness score depends on the fake news algorithm; therefore, we present an approach based on the token curated registry (TCR) to help the newsreaders further challenge the effectiveness of the algorithm, if they consider the genuineness score to be incorrect (see Section IV-D).

C. WhistleBlower Core

The key functionalities are executed inside the WhistleBlower core, which consists of *a blockchain or a DLT platform, decentralized storage, and a collection of computation nodes, either playing the role of a solver or a verifier*.

1) Role of Blockchain: WhistleBlower requires a blockchain or a DLT platform with support for smart contracts, cryptocurrency, and identity and reputation management. The *smart contracts* are used for orchestrating the fake news detection process (discussed in Section IV-C2). The *Cryptocurrency* support is desired to reward the community members for their contributions to WhistleBlower - note that the community members are engaging either by providing computation resources or through the TCR. Lastly, the identity and reputation management feature is necessary to keep track of the nodes, users, and algorithm developers' identity.

2) Smart Contract: The fake news detection application is deployed as a *smart contract*, which maintains the list of solver and verifier nodes and the pointers to the software that should be executed to derive the genuineness score. Whenever a news item is received from a WhistleBlower client, the smart contract selects two random nodes from the pool of solver nodes. It dispatches the news item along with pointers to the software for the execution.

3) Verifiable Computation using Solver and Verifier Nodes: The verifiable computation paradigm enables the application or task owner ("submitter") to off-load computation tasks to remote compute nodes ("solver" or "worker") to either speed up the computation by using powerful nodes or leverage the cheap compute nodes. When the worker nodes successfully perform the computation, the results must be verified by "approver" or "verifier" nodes. To verify the result, the "approvers" need not redo the entire computation since that would minimize the effectiveness of the computation-off-loading approach.

WhistleBlower's core employs a verifiable computation framework to involve community members in the fake news detection process, wherein the community members may contribute "solver" and "verifier" nodes. It works as follows:

1) the *solver* nodes receive computation jobs (computation software package) from the WhistleBlower smart contract. In particular, the computation jobs are assigned to two randomly selected worker nodes - $\mathcal{N}_\mathcal{A}$ and $\mathcal{N}_\mathcal{B}$.
2) When the worker nodes, $\mathcal{N}_\mathcal{A}$ and $\mathcal{N}_\mathcal{B}$, run the computation, they are required to collect evidence by capturing execution traces including stack and run-time traces, which are then submitted back to the smart contract along with the results of the computation, in the form of a pointer to the verification software package.
3) Subsequently, WhistleBlower's smart contract selects a random node to verify the correctness. The verification process involves comparing the execution traces and the output submitted by the *solver* nodes. If they match, then the computation is considered correct. If not, the WhistleBlower penalizes *both the nodes* and restart the entire process.

Our verifiable computation framework makes the following assumptions: a) Nodes are not colluding, and the random selection of nodes leads to the scheduling of jobs on two completely random nodes. b) There is no private channel between the worker nodes to copy the results to deceive WhistleBlower. Remember that the private channel let one node perform the computation while allowing the other node to copy the results and the execution traces, and present the result to the WhistleBlower as if it has executed the code.

4) Decentralized Storage: WhistleBlower considers decentralized storage solutions such as IPFS and Sia for storing

ML/AI algorithms, along with trained models and data sets. Remember contemporary blockchain and DLT platforms are not good for storing hundreds of MBs of data.

D. Token curated registry (TCR)

The token curated registry is a decentralized, community-oriented, voting-based, and incentive-driven platform for content curation. In particular, it is used to curate lists by leveraging blockchain's smart contract and tokenization capabilities. TCR works following the "Wisdom of the Crowd" idea. The group of community members is believed to be effective in making a smart decision collectively, instead of relying on a single expert. AdChain [15] is an example of a TCR application, which was created to curate a list of reliable advertisement publishers. TCR is also considered for reviewing and publishing of academic journals [16].

TCR allows the community members to challenge any item, $\mathcal{I}$, in the list by depositing tokens, which starts the challenge period, during which the community members either vote in favor or against the challenge. Here, the term "challenge" refers to a process wherein a community member questions the presence of a particular item in the list and its position. In some cases, a community member may initiate a challenge when a specific item does not deserve to be at the number one position in the list.

At the end of the challenge period, if the challenger wins (which means, the majority of the community members voted in favor of the challenger), then part of the deposit made by the owner of the $\mathcal{I}$ would be given to the challenger and $\mathcal{I}$ is removed from the list. Otherwise, the part of the challenger's deposit is given to the owner of the item $\mathcal{I}$. At the end of the challenge period, the TCR evenly distribute the part of either challenger's (if challenger loses) or item owner's (if the challenger wins) tokens to the voters that ended up on the winning side. WhistleBlower employs TCR to engage the community members in the fake news detection process.

Recall from Section III that there are various ML and AI algorithms for detecting fake news. We describe how TCR can be used to curate the fake news detection algorithms to improve the algorithms' accuracy while providing an opportunity to retrain the models through a community-driven feedback loop, as shown in Figure 1. And it works as follows:

1) WhistleBlower invites the fake news algorithm developers to post their algorithm to the "FakeNewsTCR", which maintains the curated list of fake news detection algorithms.
2) The news consumers and the algorithm developers must buy tokens from the "FakeNewsTCR" platform for algorithm registration, challenging, and voting processes.
3) Algorithm developers can submit their algorithms to the "FakeNewsTCR" list by depositing their tokens.
4) When community members (or clients) submit a news item to the smart contract for fake news assessment (see Section IV-B), they can select an algorithm from the list or let the smart contract choose the top algorithm from the list. WhistleBlower returns the genuiness score after assessing the news item using the verifiable computation framework. The client can then post the genuiness score as a sub-tweet or comment to the news item.
5) When news readers think any news item is reporting fake news and the genuineness score does not reflect it. They can check the algorithm used to compute the score. They can then go to the "FakeNewsTCR" list to challenge the algorithm. During the challenge period, they can provide evidence to the voters by presenting additional facts from credible sources. For example, AdChain provided a Reddit like a discussion platform for people to debate during the challenging phase.
6) After the challenge period ends, if a given algorithm is found to be ineffective in detecting fake news, it will be removed from WhistleBlower. The part of the algorithm developer's deposit would go to the community member who created the challenge. Simultaneously, the remaining portions of tokens would be shared among the voters who voted in favor of the challenge. We believe this model would allow the algorithm developers to develop high-quality algorithms.
7) However, if the algorithm is found to be effective, then the challenger would lose his/her tokens, which would then be shared among the algorithm developer and voters who rejected the challenge (or disagreed with the challenger).

The TCR scheme solely relies on the token holders participating in the voting process [17]. When token holders are not genuinely engaged in fear of losing their tokens, the algorithms won't be curated correctly. Therefore, we consider TCR with the inflationary mechanism based on the work described in [17]. Following the inflationary mechanism, the engaged voters are rewarded with additional tokens for their active participation in the curation process. In contrast, the traditional TCR only distributes the tokens among the voters in the winning pool. We believe that additional rewards would further encourage the community members to actively curate the fake news detection algorithms instead of holding onto their tokens, expecting an increase in the token value. In addition, we also want to note that TCR and voting approaches face challenges, including equilibrium selection issues [18]. Therefore, it is important to carefully design the mechanism and the incentives for the content curation.

Through the TCR-driven algorithm curation, WhistleBlower allows the community members to challenge the fake news detection algorithms, which helps the algorithm developers gather feedback. As part of the challenging process, we could also let the community members share evidence and data sets to improve the algorithm developers to retrain the model to enhance accuracy. WhistleBlower is the first community-driven fake news detection platform that offers the potential to help the community members mitigate the negative societal impacts of fake news effectively.

E. Attack Vectors and Solutions

Attack 1. **Selfish algorithm developers challenging other algorithms.** A selfish algorithm developer can challenge the genuineness score calculated by other algorithms, thereby, potentially causing other algorithms to be removed from the TCR.

Solution: When the TCR contains a large number of voters, the selfish algorithm developer cannot win the challenge unless the algorithm that is under challenge is legitimately inefficient in determining the genuineness score. Besides, the commercial TCR deployments such as the AdRegistry provide a forum for the voters to discuss by sharing evidence during the challenging phase, which means the selfish algorithm developer needs to submit significant evidence to win the challenge. On the flip side, the continuous scrutiny by the other algorithm developers and the newsreaders in the community would encourage the developers to enhance their algorithms to prevent them from losing a challenge.

Attack 2. **Collusion among solvers and verifiers.** WhistleBlower relies on public nodes to calculate the genuineness score. When verifiers collude with the solvers, they can submit the false genuineness score, thereby ruining the reputation of WhistleBlower.

Solution: We assume that the verifiable computation platform elects random nodes from the list of nodes, and such randomness minimizes the chances of colluding nodes being selected for performing the computation on a given round. Besides, we consider no private communication channels exist between the verifier and solver nodes outside WhistleBlower. Assuming there was collusion that resulted in the wrong genuineness score for a news item, the newsreaders have an opportunity to challenge the score through the TCR algorithm. However, a newsreader challenges the score based on the algorithm that was used for the computation, not because the verifiable computation platform incorrectly handled the computation. Under this circumstance, an algorithm developer may get penalized for the mistake of the verifiable computation platform. To prevent such issues from happening, WhistleBlower assumes that the verifiable computation platform is ensuring correctness. In our future work, we will restrict this assumption and propose an alternative solution.

Attack 3. **Fake news creator challenging the correct genuineness score.** WhistleBlower allows the newsreaders to challenge news articles' genuineness score, which would permit even the fake news creators to challenge the legitimate score calculated by the right algorithm.

Solution: The solution for this attack is similar to the solution for Attack 1, wherein a fake news creator needs to justify the voters with strong evidence that the score is incorrect. Sensible voters who vote based on the factual evidence would prevent the fake news creator from winning the challenge. Besides, the fake news creator is penalized when he/she loses a challenge. Therefore, the fake news creator would not be able to win a challenge without taking financial risks. Remember that the fake news creator would use up all his/her tokens after losing a few challenges, which would eventually prevent him/her from starting new challenges for a fake news item, unless he/she spends more money to gain tokens.

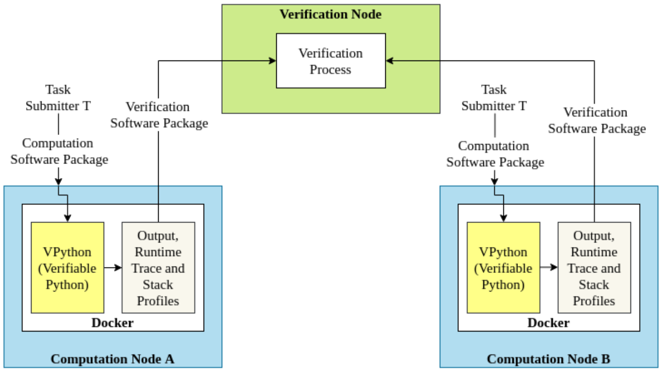
Verification Node
Verification Process
Task Submitter T
Verification Software Package
Computation Software Package
VPython (Verifiable Python)
Output, Runtime Trace and Stack Profiles
Docker
Computation Node A
Task Submitter T
Verification Software Package
Computation Software Package
VPython (Verifiable Python)
Output, Runtime Trace and Stack Profiles
Docker
Computation Node B

Fig. 2. Overview of Verifiable Python Evaluation Setup.

V. Implementation and Evaluation

A. Preliminary Implementation and Evaluation of Verifiable Python

WhistleBlower relies on a verifiable computation framework to reliably off-load computation to public nodes. We have developed a new version of Python, which we call as Verifiable Python (or vPython) by extending Python (Python-3.8) with support for collecting run-time and stack traces. In particular, we have modified the Python interpreter and runtime to gather stack and run-time traces. For this paper, the evaluation used the preliminary version of VPython, but we plan to extend the implementation with more user-friendly software interfaces and release it as open-source software. The version used for the evaluation is currently available in GitHub: https://github.com/ANRGUSC/vPython.

Evaluation Setup: Figure 3 shows the evaluation set up. Our goal is to understand the effectiveness and the performance of vPython for verifiable computation. Therefore, we only focused on the vPython components, and we did not use smart contracts for node selection, as described in Section IV-C.

Example Applications: To understand the performance overhead of vPython, we have used three example applications: **Add:** This application consists of four-lines of Python code, which define two variables, perform addition, and then print the result.**Sub:** This application is similar to Add, and it consist of a four-lines of Python code, which defines two variables, performs subtraction, and then print the result. **Fake news detection:** We have used an open-source fake news detection application from https://github.com/nishitpatel01/Fake_News_Detection, and executed it using vPython. This application takes the news headline as an input string and then output the genuineness as a result. This code uses a model that was created using the LIAR dataset [19].

Table I shows the evaluation results from executing the example applications on two distinct computation nodes. The second and third columns in Table I lists the sizes of the computation and verification software packages. We observed

Application	Computation Software Package	Verification Software Package	Total Number of Library Files Used	Total Number of Functions Used	Similarity of Runtime Traces	Similarity of Output
Add	2KB	2MB	32	252	100%	100%
Sub	2KB	2MB	32	252	100%	100%
Fake News Detection	19094KB	21MB	856	2921	100%	100%

TABLE I

vPython Evaluation Results. The similarity scores are calculated by comparing runtime traces collected from two distinctive compute nodes. The score of 100% denotes that both the nodes executed the same files and functions, which proves that both the nodes actually executed the code.

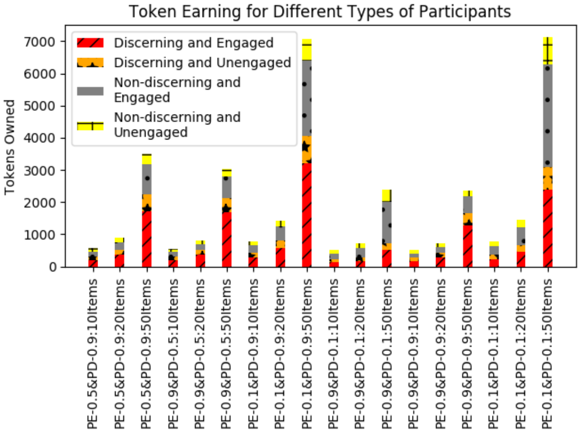

Fig. 3. Token Earning Depends on Discernment and Engagement.

that the sizes of the traces were consistent on both machines. The package size for the verification package is larger because vPython collects run-time and stack traces when the code executes on the compute node, which is then returned to the verification node, as shown in Figure 3.

The fourth and fifth columns in Table I shows the number of files and functions (within each file) that are invoked by Python interpreter during the code execution. A simple add and subtract are not relying on many libraries/modules. In contrast, a moderately complex fake news detection application uses 856 files and 2921 functions to estimate the genuineness score for a news headline. These numbers show that when a compute node runs a particular computation, it can't produce a correct result without invoking the required number of libraries and modules. The sixth column in Table I shows that the traces from two computation nodes match with 100% accuracy. In a realistic setting, this feature of vPython provides a reliable method to verify the computation's correctness, as long as the computation nodes are chosen randomly, and there are no malicious computation nodes in the network with private channels. Note that when computation nodes have private channels, only one node can run the result and share the traces with the other node. We will devise methods to overcome this limitation in our future work.

These results show that the verifiable python can provide run-time traces to verify the correctness of computations. Besides, vPython allows the application developers to run any arbitrary Python code without making any changes to their code, which we believe is one of the most significant advantages of vPython. Recall that TrueBit [20] also enables support for gathering run-time traces, but it requires the software to be written Web Assembly (WASM), which lacks support for widely used ML and AI algorithms.

B. Assessing the Effectiveness of Token-curated Registry

WhistleBlower's effectiveness and wide-spread adoption depend on its accurate estimation of genuineness score, which is calculated by the fake news detection algorithms submitted by the developers. When the newsreaders do not challenge the algorithms that are listed in the WhistleBlower registry, then WhistleBlower would be ineffective. Therefore, community participation through the TCR is critical for the success of WhistleBlower.

To study how the different types of community members influence WhistleBlower, we have used TCRSim [17], an open-source simulator for studying the effectiveness of TCR. Unlike traditional TCR, WhistleBlower incentivizes the active participants more for their engagement with the TCR. Our evaluation focuses on the following question: *what types of community members can benefit from WhistleBlower while receiving a significant amount of tokens as a reward?* We classify the participants as "discerning" and "engaging", wherein the discerning participants have the ability to dissect the facts in the news item, while engaging participants have the strong desire to actively participate in the voting process. These classifications lead to the following four categories: **Discerning-engaged**, **Discerning-Unengaged**, **Non-discerning-engaged**, and **Non-discerning and Unengaged**. Our evaluation estimates how the participants in the above categories gain tokens by running multiple simulations following different settings for the probability of engaged (PE) and the probability of being discerned (PD). To study how the number of algorithms in TCR influences the tokens gained, we have also considered 10, 20, and 50 algorithms (or items in the list). Figure 3 shows our evaluation results. "PE-x&PD-y-nItems" in the x-axis represent the number of tokens owned by different types of participants when the probability of engagement is x, probability of being discerned is y and the number of items (or algorithms) in the list in n. All the participants own 100 tokens at the start, of our evaluations, and we assume that the participants voting behavior do not change over time. The key findings are as follows:

- The higher number of items in the TCR listing increases the opportunity to earn more tokens.
- Active participation increases the chances of earning more tokens, and the reward increases further if the participant is informed.
- The participants that neither participate nor have the knowledge (i.e., discerned) do not benefit much from merely buying and owning tokens.
- The knowledgeable community members with subject knowledge do not benefit if they do not participate in the algorithm curation process.

WhistleBlower requires a high proportion of active and informed participants to reliably curate the algorithms while receiving an incentive to enhance the quality of the algorithms, which, in turn, would enhance the accuracy of the genuineness score. Therefore, WhistleBlower must either set aside tokens for rewarding participation or mint tokens on the fly. Besides, the TCR designs must carefully study the incentives and payoffs to achieve a desired equilibrium[3] [18].

C. Ongoing Work

There is an ongoing work that intends to use HelixNetwork, which comes with HelixMesh, a double consensus protocol, and a native HLX coin. It consists of an on-chain and off-chain consensus model, which offers flexibility. Besides, we are also implementing the node selection algorithm for selecting solver and verifier nodes. And, the incentives for the participants will also be described in our future work.

VI. Conclusion

Social networking sites and digital media platforms are infested with fake news, starting to harm society. We have presented WhistleBlower, a decentralized fake news detection platform by combining machine learning/AI algorithm, blockchain technology, verifiable computation framework, and token-curated registry. Besides, we have also shown how WhistleBlower helps the machine learning and AI algorithm developers to apply their solutions to a community-driven fake news detection platform. A verifiable computation framework has been used for executing the computation on public nodes contributed by the community members. And, we have shown how Token-curated registry can be used for curating the ML/AI algorithms. Our preliminary implementation and evaluation show the effectiveness of a Python-based verifiable computation platform (vPython). Lastly, we have also validated the importance of community engagement for maintaining the quality of fake news detection algorithms.

Acknowledgment

This work was supported by the USC Viterbi Center for Cyber-Physical Systems and the Internet of Things (CCI).

[3] https://medium.com/prysmeconomics/nash-equilibrium-and-blockchain-d6a6f47a7a37

References

[1] E. C. T. Jr., Z. W. Lim, and R. Ling, "Defining "fake news"," *Digital Journalism*, vol. 6, no. 2, pp. 137–153, 2018.

[2] N. Grinberg, K. Joseph, L. Friedland, B. Swire-Thompson, and D. Lazer, "Fake news on twitter during the 2016 U.S. presidential election," *Science*, vol. 363, no. 6425, pp. 374–378, 2019.

[3] K. Sharma, S. Seo, C. Meng, S. Rambhatla, A. Dua, and Y. Liu, "Coronavirus on social media: Analyzing misinformation in twitter conversations," *ArXiv*, vol. abs/2003.12309, 2020.

[4] N. Ruchansky, S. Seo, and Y. Liu, "Csi: A hybrid deep model for fake news detection," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 797–806.

[5] J. Kim, B. Tabibian, A. Oh, B. Schölkopf, and M. Gomez-Rodriguez, "Leveraging the crowd to detect and reduce the spread of fake news and misinformation," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, ser. WSDM '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 324–332.

[6] Y. Wang, F. Ma, Z. Jin, Y. Yuan, G. Xun, K. Jha, L. Su, and J. Gao, "Eann: Event adversarial neural networks for multi-modal fake news detection," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, ser. KDD '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 849–857.

[7] A. Qayyum, J. Qadir, M. U. Janjua, and F. Sher, "Using blockchain to rein in the new post-truth world and check the spread of fake news," *IT Professional*, vol. 21, no. 4, pp. 16–24, 2019.

[8] I. S. Ochoa, G. de Mello, L. A. Silva, A. J. P. Gomes, A. M. R. Fernandes, and V. R. Q. Leithardt, "Fakechain: A blockchain architecture to ensure trust in social media networks," in *Quality of Information and Communications Technology*, M. Piattini, P. Rupino da Cunha, I. García Rodríguez de Guzmán, and R. Pérez-Castillo, Eds. Cham: Springer International Publishing, 2019, pp. 105–118.

[9] A. Dorri, S. S. Kanhere, and R. Jurdak, "Towards an optimized blockchain for iot," in *2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*. IEEE, 2017, pp. 173–178.

[10] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for iot security and privacy: The case study of a smart home," in *2017 IEEE international conference on pervasive computing and communications workshops (PerCom workshops)*. IEEE, 2017, pp. 618–623.

[11] D. M. J. Lazer, M. A. Baum, Y. Benkler, A. J. Berinsky, K. M. Greenhill, F. Menczer, M. J. Metzger, B. Nyhan, G. Pennycook, D. Rothschild, M. Schudson, S. A. Sloman, C. R. Sunstein, E. A. Thorson, D. J. Watts, and J. L. Zittrain, "The science of fake news," *Science*, vol. 359, no. 6380, pp. 1094–1096, 2018.

[12] X. Zhou and R. Zafarani, "Fake news: A survey of research, detection methods, and opportunities," *ArXiv*, vol. abs/1812.00315, 2018.

[13] C. Shao, G. L. Ciampaglia, O. Varol, A. Flammini, and F. Menczer, "The spread of fake news by social bots," *arXiv preprint arXiv:1707.07592*, vol. 96, p. 104, 2017.

[14] N. Hassan, G. Zhang, F. Arslan, J. Caraballo, D. Jimenez, S. Gawsane, S. Hasan, M. Joseph, A. Kulkarni, A. K. Nayak, V. Sable, C. Li, and M. Tremayne, "Claimbuster: The first-ever end-to-end fact-checking system," *Proc. VLDB Endow.*, vol. 10, no. 12, p. 1945–1948, Aug. 2017.

[15] M. Goldin, A. Soleimani, and J. Young, "The adchain registry," *Technical White Paper*, 2017.

[16] A. Kosmarski and N. Gordiychuk, "Token-curated registry in a scholarly journal: Can blockchain support journal communities?" *Learned Publishing*, vol. 33, no. 3, pp. 333–339, 2020.

[17] Y. L. Wang and B. Krishnamachari, "Enhancing engagement in token-curated registries via an inflationary mechanism," in *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2019, pp. 188–191.

[18] A. Asgaonkar and B. Krishnamachari, "Token curated registries - a game theoretic approach," *ArXiv*, vol. abs/1809.01756, 2018.

[19] W. Y. Wang, "" liar, liar pants on fire": A new benchmark dataset for fake news detection," *arXiv preprint arXiv:1705.00648*, 2017.

[20] J. Teutsch and C. Reitwießner, "Truebit: a scalable verification solution for blockchains," 2018.